AMENDMENT TO THE CLAIMS

Please **AMEND** claims 1, 3, 5, 7, 8,14, 16 and, 24, 25 as follows.
Please **CANCEL** claims 2 and 15.

A copy of all pending claims and a status of the claims are provided below.

1. (currently amended) A method of generating cyclic redundancy checks (CRCs) for a message with N data blocks, comprising:

calculating a partial CRC for an out of order data block and storing the result;

generating a CRC remainder multiplier associated with the out of order data block and storing the result;

repeating the calculating and generating steps until all N data blocks for the message are received; ~~and~~

combining the results of the calculating step and the generating step; and

calculating a CRC for an in order data block using any previously computed in order CRC.

2. (cancel)

3. (currently amended) The method of claim [[2]] 1, further comprising computing a final CRC by combining the results of the combining step and the calculating a CRC step.

4. (original) The method of claim 3, wherein the computing step includes a divide by a generating polynomial.

5. (currently amended) The method of claim [[2]] 1, further comprising starting a first CRC engine for calculating the CRC for the in order data block, and starting the first CRC engine and a second CRC engine for calculating the partial CRC for the out of order data block, wherein the first CRC engine and the second CRC engine are adapted to be implemented on one of a same physical hardware and a different physical hardware.

6. (original) The method of claim 5, wherein in the starting step when calculating the partial CRC for the out of order data block, the first engine computes the partial CRC and the second engine computes the CRC remainder multiplier.

7. (currently amended) The method of claim [[2]] 1, further comprising initializing a CRC engine with a CRC remainder for the in order block, the CRC remainder being a result of a prior CRC computation.

8. (currently amended) ~~The method of claim 1, further comprising~~ A method of generating cyclic redundancy checks (CRCs) for a message with N data blocks, comprising:

calculating a partial CRC for an out of order data block and storing the result;

generating a CRC remainder multiplier associated with the out of order data block and storing the result;

repeating the calculating and generating steps until all N data blocks for the message are received;

combining the results of the calculating step and the generating step; and

initializing a first CRC engine with a partial CRC remainder and a second CRC engine with the CRC remainder multiplier, the partial CRC remainder and the CRC remainder multiplier being a result of a prior partial CRC computation.

9. (original) The method of claim 8, wherein the initializing step permits data blocks from different messages to be received correctly when intermixed by the network.

10. (original) The method of claim 1, wherein the calculating step includes calculating the partial CRC according to crc_b[k] = CRC (B_k), where crc_b[k] is the partial CRC for data block B_k and B_k is the data block bit pattern of data block k.

11. (original) The method of claim 1, wherein the generating step includes generating the remainder multiplier according to crc_2[k] = CRC (2^{S_k}), where crc_2[k] is the remainder multiplier for data block k, and S_k is the bit length of data block k.

12. (original) The method of claim 11, wherein the generating the remainder multiplier step includes supplying a bit pattern of length S_k plus one bit to a CRC engine.

13. (original) The method of claim 1, wherein the N data blocks contain at least one data block of the N data blocks that is one of a different length and a same length.

14. (currently amended) An apparatus for generating cyclic redundancy checks (CRCs) for a message with N data blocks, comprising:

a component to calculate a partial CRC for an out of order data block and to store the result;

a component to generate a CRC remainder multiplier associated with the out of order data block and to store the result; ~~and~~

a component to combine the results of the of the calculated partial CRC and the generated remainder multiplier; and

a component to calculate a CRC for an in order data block using an immediately previously calculated in order CRC, if available.

15. (cancel)

16. (currently amended) The apparatus of claim [[15]] 14, wherein the component to calculate the CRC provides for initializing a CRC engine with a CRC remainder, the CRC remainder being the result of a prior CRC computation.

17. (original) The apparatus of claim 14, further comprising a component to initialize a first CRC engine with a partial CRC remainder and a second CRC engine with the CRC remainder multiplier, the partial CRC remainder and the CRC remainder multiplier being a result of a prior partial CRC computation.

18. (original) The apparatus of claim 17, wherein the component to initialize permits data blocks from different messages to be received correctly when intermixed by the network.

19. (original) The apparatus of claim 14, comprising a component to produce a final CRC by combining the output from the component to combine results of the calculated partial CRC and the generator multiplier with the output from the component to calculate a CRC for an in order data block using an immediately previously calculated in order CRC, if available.

20. (original) The apparatus of claim 19, wherein the component to produce a final CRC includes a means to divide by a generating polynomial.

21. (original) The apparatus of claim 14, wherein the component to calculate a partial CRC provides for calculating the partial CRC according to crc_b[k] = CRC (B_k), where crc_b[k] being the partial CRC for data block k and B_k being the data block bit pattern of data block k.

22. (original) The apparatus of claim 14, wherein the component to generate a remainder multiplier provides for generating the remainder multiplier according to crc_2[k] = CRC(2^{S_k}), where crc_2[k] is the remainder multiplier for data block k, and S_k is the bit length of data block k.

23. (original) The apparatus of claim 22, wherein the component to generate the remainder multiplier includes a means to supply a bit pattern of length S_k plus one bit to a CRC engine.

24. (currently amended) The apparatus of claim 14, wherein the N data blocks contain at least one data block of the N data blocks that is one of a different length and a same length.

25. (currently amended) A computer program product comprising a computer usable medium having readable program code embodied in the medium, the computer program product includes:

a first component to calculate a partial CRC for an out of order data block and storing the result;

a second component to generate a CRC remainder multiplier associated with the out of order data block and storing the result; ~~and~~

a third component to combine the results of the of the first component and the second component; and

a fourth component to calculate a CRC for an in order data block using any previously computed in order CRC.